Comparing Goat Growth over Age and Genetics

Submitted to
Professor John DeGood
CSC 315-01 Database Systems
May 4, 2024

By
Milian Ingco, Prisha Priyadarshini, Aiden Turner

1. Research Problem

For our database project, we wanted to calculate the growth curve for a specific herd of goats. The reason why we wanted to do this is because it is hard to quantify and compare the rate of growth between different groups of goats and it is also difficult to identify statistical significance in different rates of growth without a proper statistical model/database. Not only that, it is difficult to find good dams to breed, and using our database will help the stakeholders with several problems such as: finding which dams are good to breed, what are good practices to help the dams gain weight, what goats aren't gaining weight, etc. The data that we needed to solve this problem were the live weight of the kidd, the average weight gain, birth date, tag number, gender, dam's tag number, and previous health complications for the dam. The data generated from our database will answer questions such as: how does the growth of different birth cohorts compare to each other using that question, the data will also answer other questions such as, how were the goats raised in the birth cohort compared to a different cohort?

2. Project Approach

In order to solve these problems, we began by figuring out the necessary data that would be needed for a Minimum Viable Product (MVP). Some of the data we would gather for the database were the weight of the goats, the birth date, gender, and the goats' tag numbers among others. Additionally, we figured out what our use cases would be. These would be our solution to our first and second problems. Mainly, we wanted to be able to generate quantitative data that would be able to objectively determine if there was any statistically significant difference

between the growth of goats grouped by certain criteria. The criteria we selected were based on the problems we identified. There was no easy way to visualize the growth of large groups of goats and compare newer goats to older goats or different dams' lineage to another. After identifying the data we would need, along with how we wanted to use it, we created our ER diagrams and our database's schema. However, we were initially intending on only using one relation that would just hold all of a goat's data, including the goat's weighing history. Eventually we realized that since it wasn't normalized, we would need to have separate relations for goat data, and weight growth.

After normalizing our database, we were able to create views that would be used for grouping our goats by different categories. SQL was used to create and load temporary tables used for testing that had all the essential attributes for our MVP. By using the test data, we were able to select goats that were born between different dates, and use recursion to select goats that were related to each other. Additionally, out of mostly pure curiosity we created a view that would select goats dependent on their current status being dead, sold, living, or just all goats without any grouping.

Since most of the setup was done, the task of getting the data into a format where we could graph and analyze the data wasn't too difficult since it was the same for all groups, just using different criteria. The basic way that it worked was, for some given criteria whether a dam or birth range, return a group of goats. For each of those goats, return their weighing history from an attribute table. Lastly, for each entry, get the age at the time it was weighed which will be the x component, and the weight at that age is the y component. After that is completed, graphing each of the points is trivial.

In more context, plain HTML was used to get user input and redirected it to corresponding Flask functions in Python. These functions use session data to pass the list of inputs to a separate function that parses all the data into a 4D list. That function uses the library psycopg2 to run SQL queries as formatted strings with the inputs as criteria for the SQL views I created earlier. After formatting the list, I used Plotly to create interactive scatter plots that would be shown on the website. Lastly, on the graph page we use HTML once again to select a type of regression model, for which the library statsmodel is used to generate either a linear regression or logarithmic regression. Also, for ease of use, Python automatically loads the database with data from CSV files and creates the associated tables. Due to the concepts overlapping, a large amount of work was able to be abstracted away from the specifics and done en masse.

3. Project Benefits

Our project created a detailed database system for Silvies Valley Ranch to help improve goat farming and make it more sustainable. This system gathers all sorts of important data about the goats, such as their growth, health, and genetics, and puts it in one easy-to-access place. This makes it simpler for the ranch managers to get the information they need quickly and make good decisions based on that data. The database also lets them keep an eye on how different groups of goats grow over time, which helps in planning and using resources wisely.

Additionally, the database tracks health issues and growth patterns, which helps prevent small problems from becoming big ones. It's also useful for research, helping to pick out the best breeding practices and healthier goat breeds. This not only makes the ranch run better but also

cuts down on the need for medical treatments and reduces the environmental impact. Overall, the new database is a great tool for the ranch, making day-to-day operations smoother and helping to keep things sustainable for the future.